**Alarm Clock si Stopwatch in Arduino**

Giesswein Alexia

Grupa 30231

Prof. Coord. Itu Razvan

**Cuprins**

[1. Scopul proiectului 3](file:///C:\Users\Alexia\Downloads\Documentatie_Parking_Sensors_Lupu.docx#_Toc124645129)

[2. Implementare 3](file:///C:\Users\Alexia\Downloads\Documentatie_Parking_Sensors_Lupu.docx#_Toc124645130)

[3. Circuite 6](file:///C:\Users\Alexia\Downloads\Documentatie_Parking_Sensors_Lupu.docx#_Toc124645131)

[4. Poza sistemului cablat 8](file:///C:\Users\Alexia\Downloads\Documentatie_Parking_Sensors_Lupu.docx#_Toc124645132)

[5. Bibliografie 8](file:///C:\Users\Alexia\Downloads\Documentatie_Parking_Sensors_Lupu.docx#_Toc124645133)

**Scopul proiectului**

Proiectul contine 2 programe, unul care simuleaza un ceas cu alarma si unul care simuleaza un cronometru.

Ceasul cu alarma afiseaza ora curenta si are posibilitatea de a seta o alarma. Cand ora curenta va fi egala cu ora la care este setata alarma, aceasta va suna simuland un ceas cu alarma care este folosit zi de zi. De asemenea, avem posibilitatea de a vedea ora la care este setata alarma.

Cronometrul are 2 butoane, unul de start si unul de stop. Cand se apasa pe butonul start, se incepe o numaratore, in secunde, simuland un cronometru folosit in viata de zi cu zi. Cand se apasa pe butonul stop, numaratoare se opreste si se afiseaza cate secunde au trecut de la pornirea cronometrului. Daca se apasa din nou pe start, cronometrul isi incepe numaratoarea de la inceput.

**Implementare**

Pentru realizarea ceasului cu alarma avem nevoie de urmatoarele componente: o placuta Arduino (eu am folosit Arduino UNO), un breadboard, un LCD pe care se afiseaza ora curenta si ora la care este setata alarma (eu am folosit un LCD I2C), patru butoane, un buzzer si fire mama-tata si tata-tata.

La inceput ora curenta si ora pentru alarma sunt setate la “00:00”. Primul buton este folosit pentru a seta alarma, al doilea buton pentru a seta ora de inceput, al treilea buton este pentru a modifica oreel, iar al patrulea buton pentru a modifica minutele. Pentru a seta ora se apasa butonul 3 sau 4, in timp ce se tine apasat butonul 2, iar pentru a seta alarma se apasa butonul 3 sau 4, in timp ce se tine apasat butonul 1. Pe LCD apare ora curenta si ora la care este setata alarma, iar cand cele doua sunt egale, buzzerul va suna.

Am folosit niste variabile (hours, mins, alarmhours si alarmmins) care se incrementeaza cand se apasa pe buton, pentru a seta ora sau alarma si am verificat pentru fiecare conditiile necesare: cand se trece de ora 23, ora redevine 00, cand se trece de minutul 59 ora creste iar minutele redevin 00, iar la afisare daca ora sau minutele sunt mai mici decat 0, se afiseaza un 0 in fata.

if(digitalRead(11) == LOW)

{ alarmmins++; }

 else if (digitalRead(10) == LOW)

    { alarmhours++;}

  if(alarmhours < 10)

  {

    lcd.print("0");

    lcd.print(alarmhours);

  }

  else

  {

    lcd.print(alarmhours);

  }

 if(alarmmins > 59)

     {

      alarmhours++;

      alarmmins = 0;

     }

  if(alarmhours > 23)

     {

      alarmhours = 0;

     }

}

Pentru realizarea cronometrului avem nevoie de urmatoarele componente: o placuta Arduino (eu am folosit Arduino UNO), un breadboard, un LCD pe care se afiseaza timpul de cand s-a pornit cronometrul pana la oprire, in secunde (eu am folosit un LCD I2C), doua butoane, doua LED-uri, 4 rezistente si fire.

Cand se apasa pe butonul de start, cronometrul incepe sa numere in secunde cat timp trece si se aprinde ledul verde. Cand se apasa pe butonul de stop, cronometrul isi opreste numaratoarea si pe LCD se afiseaza cate secunde au trecut si se aprinde ledul rosu. Daca se apasa din nou pe butonul start, numaratoarea incepe din nou de la 0.

Am folosit doua variabile care retin cate secunde au trecut de la inceperea programului pana la oprirea acestuia. Cand se apasa butonul de start se incepe numaratoarea si se aprinde becul verde, iar cel rosu se stinge, iar cand se apasa butonul de stop se opreste numaratoarea, se aprinde becul rosu, iar cel verde se stinge.

while(digitalRead(9) == HIGH)

   {

   c = millis();

   time = (c - a) / 1000;

   lcd.print(time);

   lcd.setCursor(7,0);

   lcd.print("Seconds");

   lcd.setCursor(0,0);

   digitalWrite(2, HIGH);

   digitalWrite(3, LOW);

   }

if(digitalRead(9) == LOW)

   {

     while(digitalRead(8) == HIGH)

     {

       lcd.setCursor(0,0);

       lcd.setCursor(11,0);

       lcd.print("");

       lcd.setCursor(0,0);

          digitalWrite(2, LOW);

          digitalWrite(3, HIGH);

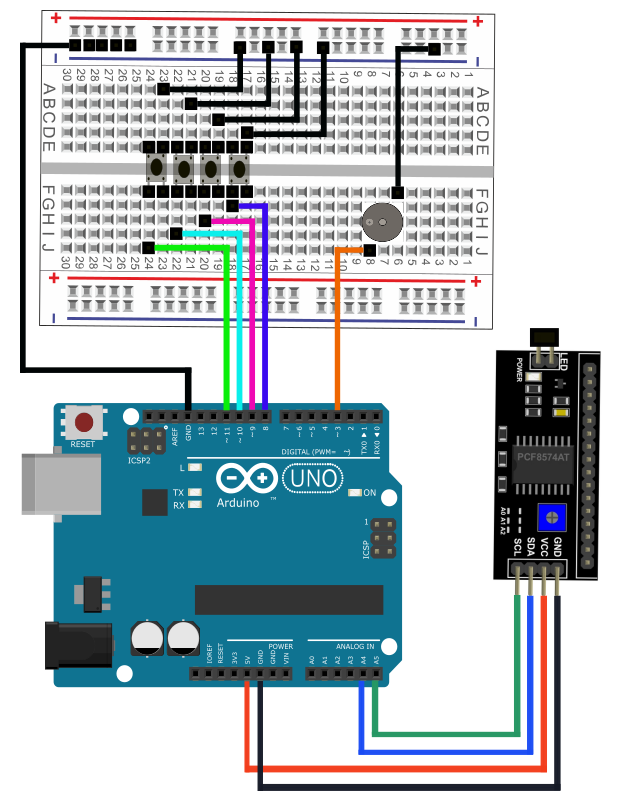
     }

   }

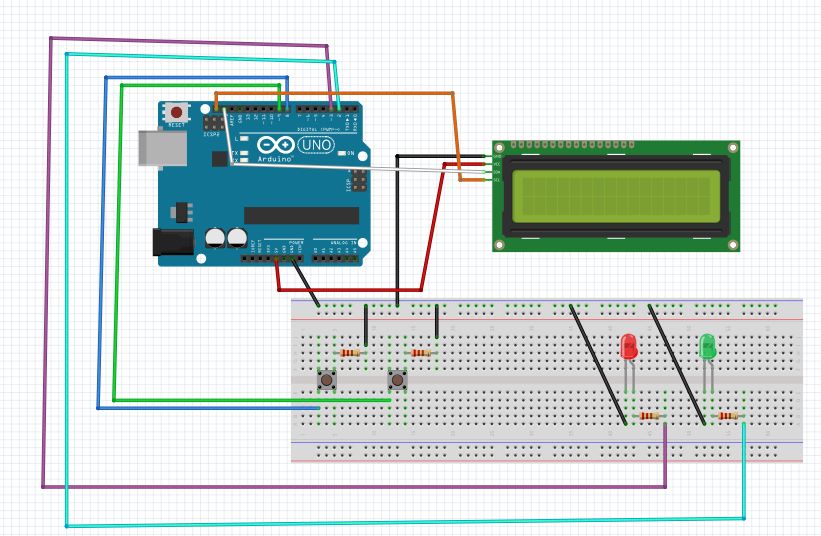
 }

**Circuite**

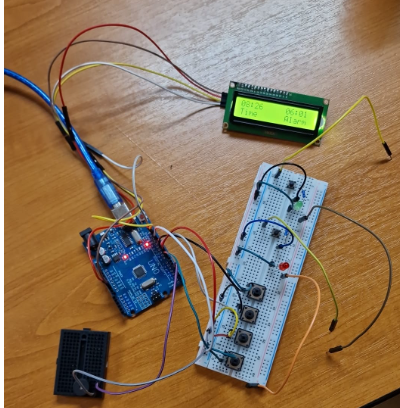
Circuitul pentru Alarm Clock:

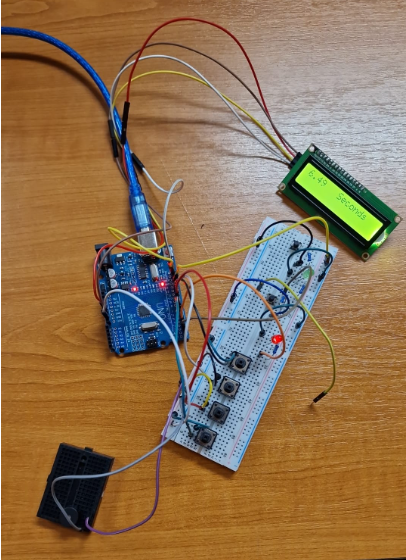


Circuitul pentru Stopwatch:



**Poza sistemului cablat:**

****

****

**Bibliografie:**

<https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/336-3.pdf>

<https://forum.fritzing.org/t/16x2-i2c-lcd-part/2041/4>